A1 of a look-up table (LUT) and/or registers of a CLB slice of the FPGA, i.e., utilizing the atomic elements of an FPGA.

Please replace the paragraph beginning on page 10, line 21 with the following:

A2 According to one implementation, summing module generator performs maximal segmentation (virtual grouping of bits denoted by dashed lines 303) within the column to group bits in groups of 3, 2, or 1 bit(s), respectively. Three-bit groups are passed to a full adder for processing, while two-bit groups are passed to a half-adder for processing. Single bit columns are passed directly to an available register 312 within a CLB. In accordance with one aspect of the invention, summing module generator 222 utilizes standard routing analysis tools to identify the optimal atomic layout of each of the allocated elements 308-312 of the hybrid Wallace tree 306. According to one implementation, summing module generator is designed to minimize waste of atomic resources and allocates elements 308-312 in this regard. According to one implementation, summing module generator 222 prioritizes performance speed over waste and, as a result, seeks to minimize routing among and between atomic elements 206-212 implementing the hybrid summing module 306, even at the expense of some waste of atomic resources. In another implementation, resource conservation and performance are equally weighted, with resources allocated accordingly.

Please replace the equation beginning on page 18, line 3 with the following:

A3 (2) $I_2=(a_2*c_2)-(b_2*d_2)$ and added to I_1; performed simultaneously with $Q_2=(a_2*d_2)+(b_2*c_2)$ and added to Q_1.

IN THE CLAIMS

Please cancel claim 1.

Please add the following new claims:

A4 2. (New) An apparatus comprising:

a plurality of inputs to receive multiple input terms;

a multi-stage series of Boolean function generators coupled with the inputs to implement one or more full-adders, half-adders, and single registers to produce intermediate summation results by combining the input terms, the series of Boolean function generators to be structured based, at least in part, on one or more attributes of the input terms; and

a multi-input adder, logically coupled with the series of Boolean function generators to produce a final sum of the input terms by combining the intermediate summation results.

3. (New) The apparatus of claim 2 wherein the input terms include one or more accumulator bits.

4. (New) The apparatus of claim 2 wherein the Boolean function generators comprise four-input look-up tables (LUTs) to implement Boolean logic functions.

5. (New) The apparatus of claim 2 wherein the Boolean function generators have associated registers.

6. (New) The apparatus of claim 2 wherein the series of Boolean function generators is structured to receive three-bit input terms by a full-adder, two-bit input terms by a half-adder, and single-bit input terms by a single register.

7. (New) The apparatus of claim 2 wherein the multi-input adder comprises an adder with an input for each single register in a final stage of the multiple stages of the series.

8. (New) The apparatus of claim 7 further comprising:

single registers in the series of Boolean function generators to receive feedback accumulator bits from the multi-input adder, the accumulator bits resulting from a multiply-accumulate operation.

9. (New) The apparatus of claim 7 further comprising:

single registers in the series of Boolean function generators to receive feedback accumulator bits resulting from a multiply-accumulate operation; and

an accumulator coupled with the multi-input adder to feed the accumulator bits back into the series of Boolean function generators.

10. (New) The apparatus of claim 2 wherein the series of Boolean function generators is incorporated in a dedicated logic device.

A4

11. (New) The apparatus of claim 10 wherein the dedicated logic device comprises a field programmable gate array (FPGA).

12. (New) The apparatus of claim 10 wherein the dedicated logic device comprises a device with control logic and a block of dedicated logic.

13. (New) The apparatus of claim 10 further comprising:

inputs to couple the dedicated logic device with a controller to structure atomic elements of the dedicated logic device into an architecture to implement the one or more full-adders, half-adders, and single registers, the architecture based, at least in part, on an analysis of the input terms.

14. (New) The apparatus of claim 10 further comprising:

a logic control module to structure atomic elements of the dedicated logic device into an architecture to implement the one or more full-adders, half-adders, and single registers, the architecture based, at least in part, on an analysis of the input terms.

15. (New) The apparatus of claim 14 wherein the analysis of the input terms comprises a bit-wise analysis.

16. (New) The apparatus of claim 14 wherein the logic control module dynamically structures the atomic elements of the dedicated logic device to implement desired instances of the architectural structure of the one or more full-adders, half-adders, and single registers.

17. (New) A method for performing complex arithmetic, comprising:

receiving a plurality of input terms;

analyzing an attribute of the input terms;

producing intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that implements one or more full-adders, half-adders, and single registers, the structure of the series of Boolean function generators based, at least in part, on the attribute of the input terms; and

producing a final sum of the input terms by combining the intermediate summation results with a multi-input adder.

18. (New) The method of claim 17 wherein receiving the plurality of input terms includes receiving one or more accumulator bits.

19. (New) The method of claim 17 wherein producing the intermediate summation results by combining the input terms with the multi-stage series of Boolean function generators comprises combining the input terms with four-input look-up tables (LUTs) that implement Boolean logic functions.

20. (New) The method of claim 17 wherein producing the intermediate summation results by combining the input terms with the multi-stage series of Boolean function generators comprises combining the input terms Boolean function generators that have associated registers.

21. (New) The method of claim 17 wherein producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators whose structure is based, at least in part, on the attribute of the input terms comprises combining the input terms with a multi-stage series of Boolean function generators structured to receive three-bit input terms by a full-adder, two-bit input terms by a half-adder, and single-bit input terms by a single register.

22. (New) The method of claim 17 wherein producing the final sum with the multi-input adder comprises producing the final sum with an adder that has an input for each single register in a final stage of the multiple stages of the series.

23. (New) The method of claim 22 further comprising:

receiving, from the multi-input adder, feedback accumulator bits resulting from a multiply-accumulate operation, with single registers in the series of Boolean function generators.

24. (New) The method of claim 22 further comprising:

receiving, from an accumulator coupled with the multi-input adder, feedback accumulator bits resulting from a multiply-accumulate operation, with single registers in the series of Boolean function generators.

25. (New) The method of claim 17 wherein producing intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators comprises producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a dedicated logic device.

26. (New) The method of claim 25 wherein producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a dedicated logic device comprises producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a field programmable gate array (FPGA).

27. (New) The method of claim 25 wherein producing the intermediate summation results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a dedicated logic device comprises producing the intermediate summation

results by combining the input terms with a multi-stage series of Boolean function generators that is incorporated in a device with control logic and a block of dedicated logic.

28. (New) The method of claim 25 further comprising:

structuring, with a controller coupled with the dedicated logic device, atomic elements of the dedicated logic device into an architecture to implement the one or more full-adders, half-adders, and single registers, the architecture based, at least in part, on an analysis of the input terms.

29. (New) The method of claim 25 further comprising:

structuring atomic elements of the dedicated logic device into an architecture to implement the one or more full-adders, half-adders, and single registers, the architecture based, at least in part, on the analysis of the input terms.

30. (New) The method of claim 29 wherein structuring the atomic elements of the dedicated logic device comprises dynamically structuring the atomic elements of the dedicated logic device to implement desired instances of the architectural structure of the one or more full-adders, half-adders, and single registers based on the analysis of the input terms.

31. (New) The method of claim 17 wherein analyzing an attribute of the input terms comprises performing a bit-wise analysis of the input terms.

32. (New) A method for performing complex arithmetic comprising:

generating a plurality of partial products from two or more input terms;

for a real-component branch, inverting certain partial products and passing the inverted and non-inverted partial products to a multi-stage series of Boolean function generators that implements one or more full-adders, half-adders, and single registers to produce intermediate summation results by combining the partial products;

for an imaginary-component branch, passing the partial products to a multi-stage series of Boolean function generators that implements one or more full-adders, half-adders, and single registers to produce intermediate summation results by combining the partial products;

receiving in both branches accumulator bits over a feedback path; and

adding the intermediate summation results with the accumulator bits for each branch to produce a final real-component sum and a final imaginary-component sum.

33. (New) The method of claim 32 wherein the method is performed on a dedicated logic device.

34. (New) The method of claim 33 wherein the method is performed on a field programmable gate array (FPGA).

35. (New) The method of claim 33 wherein the method is performed on a device having control logic and a block of dedicated logic.

36. (New) The method of claim 32 wherein generating a plurality of partial product terms comprises combining the two or more inputs in a combinatorial stage of a complex multiply accumulator.

37. (New) The method of claim 32 further comprising:

analyzing one or more attributes of the input terms.

38. (New) The method of claim 37 wherein analyzing the one or more attributes of the input terms comprises performing a bit-wise analysis of the input terms.

39. (New) The method of claim 37 wherein passing partial products to a multi-stage series of Boolean function generators that implements one or more full-adders, half-adders, and single registers further comprises structuring an architecture of the multi-stage series of Boolean function generators based, at least in part, on the analysis of the input terms.